Automated Video Capture Bird Feeder with Data Collection

ECE 445 Senior Design Document

Team #10

Kevin Li (kli56)
John Golden (jgolden4)
Colten Brunner (cbrunner)

TA: Nikhil Arora
Professor: Viktor Gruev

Spring 2024

Table of Contents

1. Introduction

1.1. Problem and Solution

Many nature enthusiasts enjoy watching birds outside their windows with homemade or store-bought feeders. This practice has been going on for many years, but until recently it has been impossible to see the birds feeding without being present. With modern-day technology, it has become possible to mount cameras onto or adjacent to bird feeders to see birds feeding, but in the new era of information technology, there should be more to bird feeders than simple footage. We seek to add onto an automated video capture system by including data capture to analyze when peak feeding hours occur. This problem occurs for common bird watchers and ornithologists alike. Whether it is knowing when to sit in front of your bird feeder or wanting to collect feeding data in specific areas, this is a problem that necessitates a solution.

The solution we propose involves a bird feeder that has a camera to turn on when motion is detected. The idea is to have an passive infrared sensor that would trigger a camera to record for a given set of times if motion is detected. In addition, specific data points that would benefit nature enthusiasts would be acquired and stored. These would include time intervals when birds arrive to identify peak bird times. We also want the end user to be able to view live footage of the bird feeder via a website URL. Our solution would implement all of this by using a power pack located on the bird feeder that supplies power to the camera, motion detector, and microcontroller that all work in tandem to create our final product.

1.2. Visual Aid

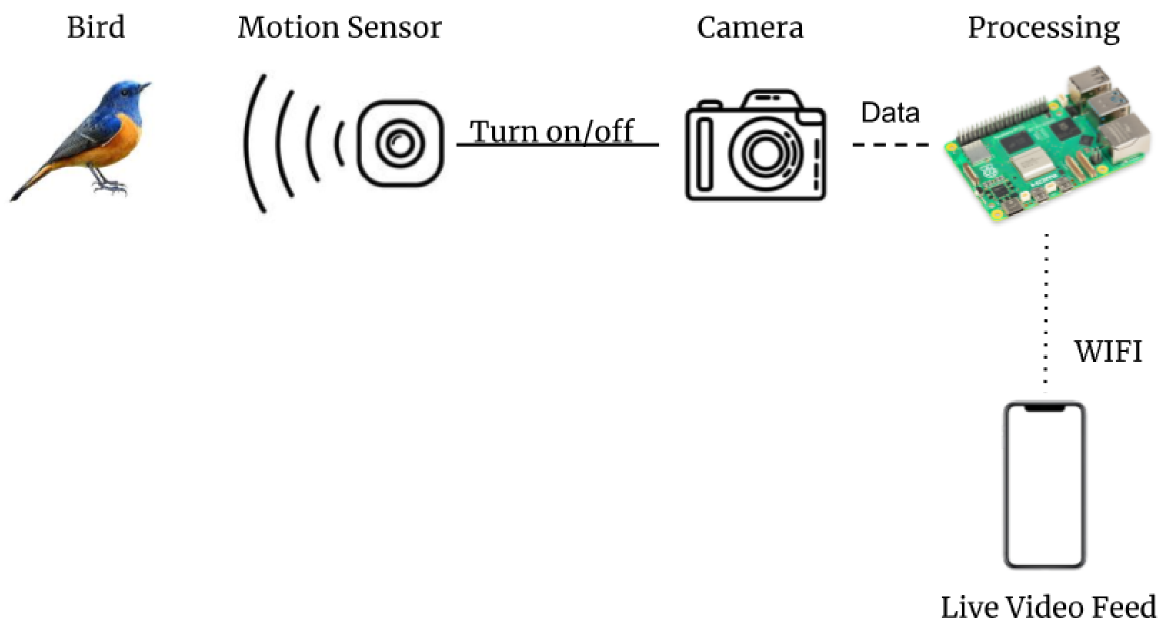


Figure 1: Visual Aid of Bird Feeder

1.3. High Level Requirements

- Activate the camera within 10 seconds of a bird landing, correctly identifying bird presence with 90% accuracy.
- Deliver a live video feed to the user device with a resolution of 720p or higher, while displaying the feed with a delay of no more than 15 seconds.
- Turn off the camera 30 seconds after the bird departs. The camera should stay on if motion is detected within the birdhouse during the timeout period, while minimizing false positives triggered by wind, leaves, or other non-bird movements.

2. Design

2.1. Block Diagram

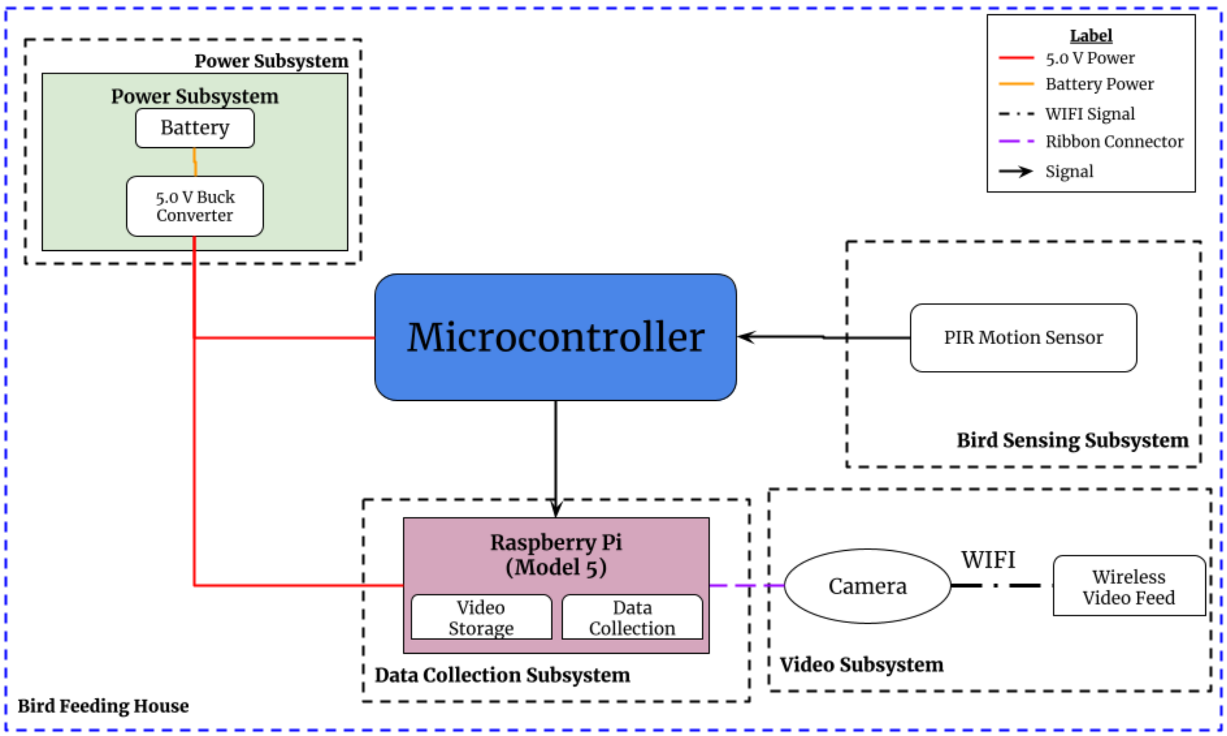
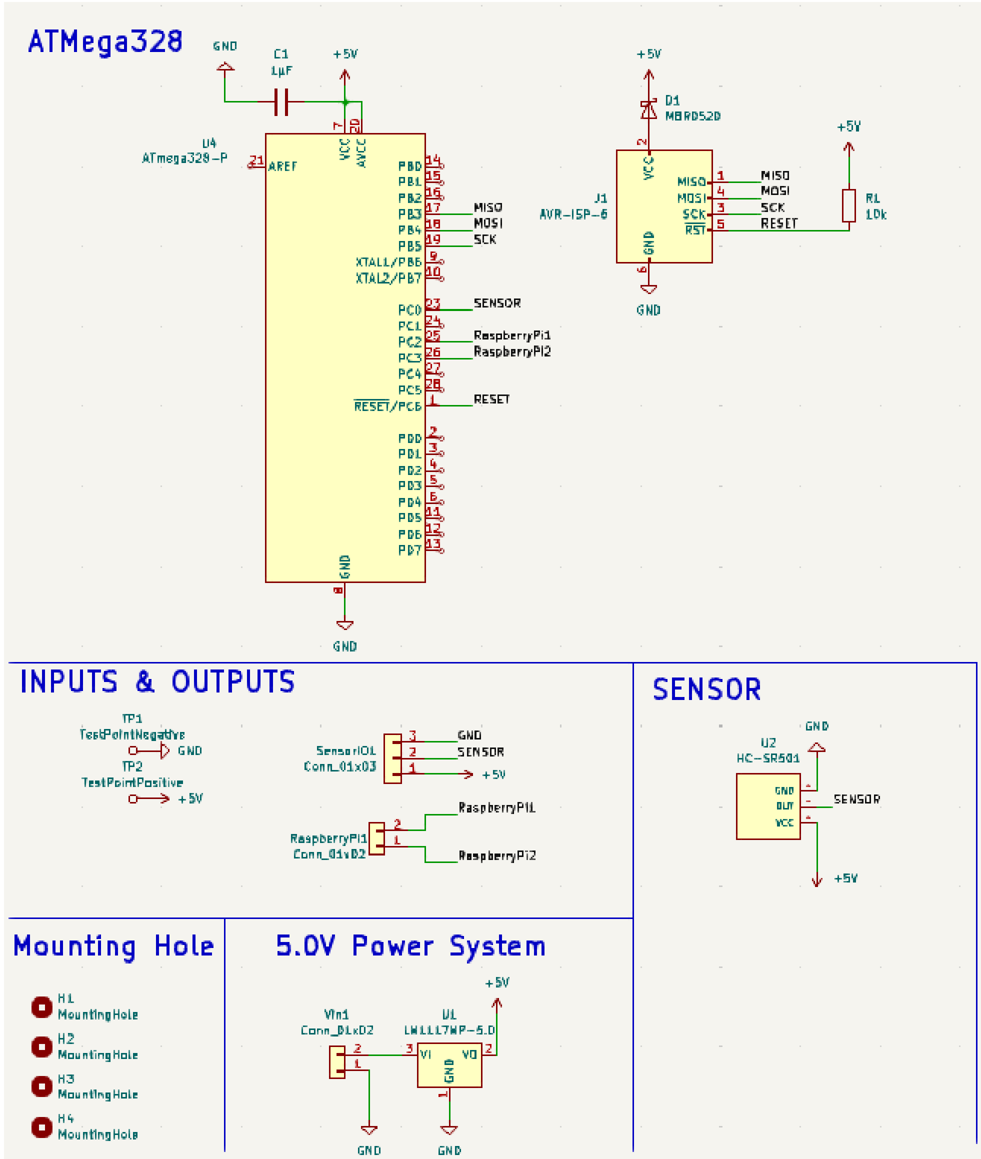
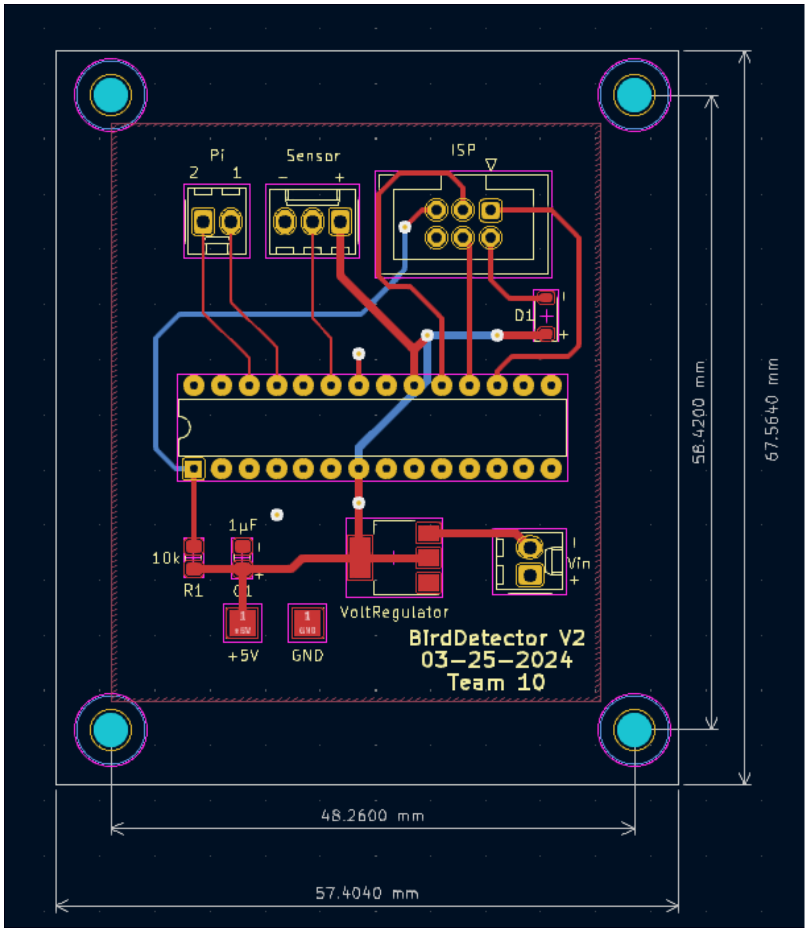


Figure 2: Block Diagram of Design


ATMega328
GND
C1
1µF
+5V
U4
ATmega328-P
AREF
VCC
AVCC
PB0
PB1
PB2
PB3
PB4
PB5
XTAL1/PB6
XTAL2/PB7
PC0
PC1
PC2
PC3
PC4
PC5
RESET/PC6
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
MISO
MOSI
SCK
SENSOR
RaspberryPi1
RaspberryPi2
RESET
GND
D1
MBRD520
J1
AVR-ISP-6
R1
10k
INPUTS & OUTPUTS
TP1
TestPointNegative
TP2
TestPointPositive
SensorIO1
Conn_01x03
RaspberryPi1
Conn_01x02
SENSOR
U2
HC-SR501
Mounting Hole
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
5.0V Power System
Vin1
Conn_01x02
U1
LM1117MP-5.0
VI
VO



2.2. Design Overview

The subsystems can be viewed as separate systems that will be implemented where some are primarily hardware focused and others software focused. Namely, the video capture, data collection and real-time video feed are mainly software connected to the raspberry pi system. Specifically the video capture will have an input from the microcontroller to turn on the camera given the infrared sensor outputs detection. Then that detection will feed into one of the raspberry pi GPIO pins which will in turn then handle all in software the rest of the video capture, data collection and real-time video feed subsystems. The hardware focused subsystems are the power system and system housing. While the system housing will also hold the raspberry pi system it is mainly used to talk about the setup of the PIR sensor. Like stated previously the PIR sensor will be connected to the microcontroller and the raspberry pi via GPIO pins. The power system will be connected mainly to the microcontroller circuit via the pcb as the raspberry pi will be on a separate wall power source due to the computer being bought personally.

2.3. Subsystem Requirements

2.3.1. Subsystem 1 - Video Capture

The subsystem, composed of a Raspberry Pi, camera, camera ribbon, and SSD, is responsible for initiating video recording upon receiving a signal from the microcontroller and storing the captured data on the SSD. We have focused on three primary aspects: timeliness of recording start, video data processing rate, and video quality. Below is the specific requirements and verification methods for each key aspect:

Requirement	Verification
Timeliness of Recording Start: **1.** Raspberry Pi should initiate recording within 10 seconds upon signal from the microcontroller.	**1.1.** Measure the time delay between signal reception and the start of video recording using a timer. **1.2.** Repeat the measurement multiple times and calculate the average and standard deviation of the delay. **1.3.** Ensure the average delay is less than 10 seconds.
Video Data Processing Rate: **2.** The Raspberry Pi processes video data at an average rate of 0.4 MB/s in SD resolution.	**2.1.** Capture a short video clip (10 seconds) and measure the file size. **2.2.** Calculate the processing rate (data size / recording time) in MB/s. **2.3.** Repeat the measurement several times with different lighting conditions and object movements. **2.4.** Ensure the average processing rate is close to 0.4 MB/s.
Video Quality: **3.** The recorded video clips should be at least 720p resolution	**3.1.** Capture a short video clip (10 seconds). **3.2.** Manually review recorded video clips for artifacts, blurriness, or excessive noise.

2.3.2. Subsystem 2 - Data Collection

The data collection subsystem focuses on collecting and storing data. The system is primarily to be tested via the software we are writing for this subsystem as the hardware systems should still be tested but involve the raspberry pi functioning correctly in tandem with the microssd associated with this device. The software involves creating CSV files on the Raspberry Pi at regular intervals, ensuring they arrive on time, containing the correct information in the proper format, and ultimately if they are securely saved on the SSD. Additionally due to further complexity requirements we will use an OpenCV model namely looking at using: https://github.com/ccrenfroe/BirdCam?tab=readme-ov-file and https://www.kaggle.com/models/google/aiy/frameworks/tfLite/variations/vision-classifier-birds-v1 for the specific model. However, if the raspberry pi can run a more robust but larger model I will look at using this model: https://www.kaggle.com/datasets/gpiosenka/100-bird-species

Below is the specific requirements and verification methods for each key aspect:

Requirement	Verification
Time of CSV Generation: **1.** The Raspberry Pi generates a new CSV file within 10 seconds of the designated time interval.	**1.1.** Record the timestamp of CSV creation. **1.2.** Calculate the time difference between the expected and actual creation times. **1.3.** Repeat the measurement multiple times and analyze the distribution of time differences.
CSV File Content: **2.** The CSV file contains all necessary data columns in the correct format.	**2.1.** Manually view the CSV file to view the data and see if the data is there.
CSV File Storage: **3.** The CSV file on the SSD has a unique and identifiable filename.	**3.1.** Manually check the filename of the generated CSV file to make sure it has the timestamp.
OpenCV: **4.** The accuracy should be within 10% of the accuracy of the model.	**4.1** Flash images of birds to test the model before implementation and create a baseline for the model **4.2** Parse images from videos of livefeed bird videos to test its drop from camera quality, etc.

2.3.3. Subsystem 3 - Power system

The power system ensures consistent voltage delivery for the raspberry pi and the microcontroller, guaranteeing smooth operation for both components. The raspberry pi requires 5.0 volts [5] and the microcontroller requires 5.0 volts. By measuring and monitoring the voltage during various operating conditions, we can verify its stability and avoid potential power-related issues. In the initial development stages, this subsystem takes power from a wall outlet and for the final product it will be using a battery. Below is the specific requirements and verification methods for each key aspect: [4]

Requirement	**Verification**
Raspberry Pi: **1.** The power system should supply dc voltage to the Raspberry Pi between 4.75 - 5.25 volts	**1.1.** Connect a voltmeter to the 5.0V voltage regulator power output **1.2.** Run the system through various operating scenarios (including idle, high load) **1.3.** Ensure the voltage stays within 4.75 - 5.25 volts
Microcontroller ATmega328p: **2.** The power system should supply dc voltage to the microcontroller be 5.0 volts ± 0.5 volts [6]	**2.1.** Connect a voltmeter to the 5.0V voltage regulator power output **2.2.** Run the system through various operating scenarios (including idle, high load) **2.3.** Ensure the voltage stays within 4.5 to 5.5 volts

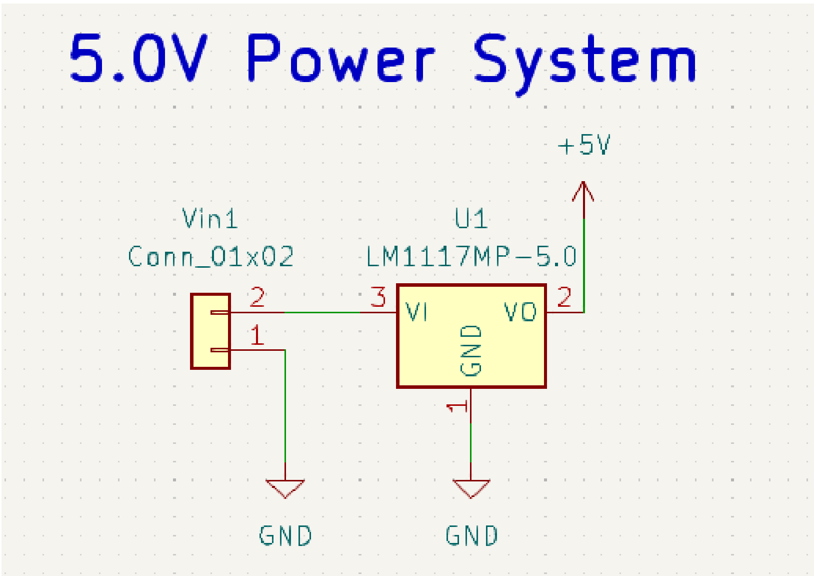


Figure 3: Power System

2.3.4. Subsystem 4 - Real-time Video Feed

This system is using the raspberry pi as a broadcasting server. Either via a web server or a direct connection a user should be able to see video streaming.

Requirement	Verification
Stream Delay: **1.** Delay between the stream and live should be within 15 seconds	**1.1.** Set up a simple test environment with the camera capturing both the live feed and recording for streaming. **1.2.** Simultaneously introduce a visual cue (e.g., hand wave) in front of the camera. **1.3.** Measure the time difference between the cue appearing in the live feed and its appearance in the streamed video.
Video Quality: **2.** The broadcast should as least be SD quality	**2.1.** Display the streamed video on a remote device capable of showing resolution details. **2.2.** Continuously monitor the displayed resolution to ensure it never drops below SD quality.
Bird Name Display: **3.** The Broadcast should have a list of the current birds identified on the feed	**3.1** If the bird can be identified check if a physical identifier ig name is displayed on the web server.

2.3.5. Subsystem 5 - System Housing

The system is enclosed in a waterproof container with two wires for the camera and ultrasonic sensor exiting the container. The wires protruding from the enclosure are waterproofed with silicone to prevent water going in. The camera is placed inside the bird feeder, facing the feed, while the ultrasonic sensor is positioned nearby. When birds land, the sensor detects motion, activating the camera for recording.

Requirement	Verification
PIR Performance in Simulated Weather **1.** Sensor does not detect motion from inanimate objects	**1.1.** Connect the PIR and place it away from living objects **1.2.** Use a tool to initiate motion from a distance while no living being is within range **1.3.** Continuously monitor the sensor output. If any motion is detected during this process, consider it a failure
PIR Protection from weather **2.** Transducer does not get damaged in severe weather	**2.1.** Visually inspect the mounted location of the transducer. Ensure it is fully sheltered from direct rain, snow, or hail. **2.2.** If necessary, adjust the mounting position or add a protective cover to shield the transducer from water exposure. **2.3.** Document the chosen protection method and its effectiveness in preventing water damage.
Environmental Protection for all electronics **3.** The enclosure housing the electronics must be weatherproof	**3.1.** Visually inspect the enclosure for proper sealing around seams, ports, and cable entries. **3.2.** Conduct a water spray test to verify the enclosure's resistance to rain and splashing water.

2.3.6. Subsystem 6 - Motion Detection

We decided to separate the sensor into a separate subsystem. This subsystem uses the HC-SR501 PIR motion detector which uses infrared to detect if a living creature is in front of the sensor. This subsystem directly integrates wtth the microcontroller using arduino code to correctly output a signal to the raspberry pi. [3].

Requirement	Verification
1. Motion sensing used to see if infrared can pick up hand	**1.1.** Put hand in front of motion sensor and check if the output from the sensor is high voltage which is 3.3V and

in front of sensor	once the hand is removed drop down to 0V. Voltage will be checked via a voltmeter.

2.4. Tolerance Analysis

One key subsystem in our project is the power for the entire project. In our power subsystem we want to ensure that the bird feeder can operate for a reasonable amount of time without a charger or battery replacement. To determine the possible lifetime from the battery pack we are using, we need to first find the total current draw from the battery pack and then compare it to the datasheet rating. Below is a table with the necessary currents from different components

Component	ATMega328	PIR Sensor	LM117
Current Draw	20 mA	~10 mA	~15 mA
Voltage	3.3 V	5 V	3.3V-5V

$$T = I_{total} * Rating = \frac{2500mAh}{(20mA+10mA+15mA)} = 55.56h$$

Using these values, we can simply multiply our total current by our batteries' 2500 mAh rating to get the predicted time the power would last well over two days. This time is more than enough for our intended use to examine birds over a weekend with minimal intervention as food would likely run out before battery life.

One more subsystem that required analysis was the passive infrared sensor. The concern for this system was that the sensor needs to capture the full area in the bird feeder so that no incident birds are missed by this system. For this calculation, the physical dimension of the feeder is needed as well as the placement area of the sensor. Shown in the figure below is the approximate geometry of our bird feeder. The PIR is rated for 120º conical range, so to ensure there are no blind spots in our viewing area, some geometrical calculations are necessary. Every angle needs to be less than 60º on either side of the PIR's orientation to fully capture the feeding area. Shown below are the 3D, top, and side views of the bird feeder. The PIR location is represented by the light blue box, the PIR orientation is in yellow, and the green line is used as an orientation reference.

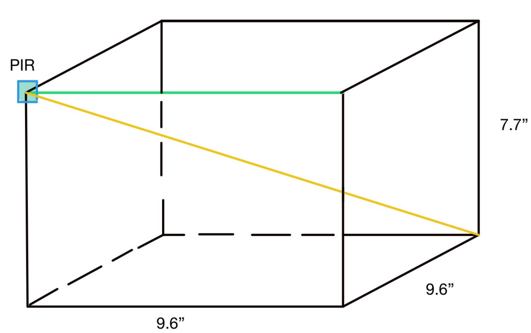


Figure 4: 3D view

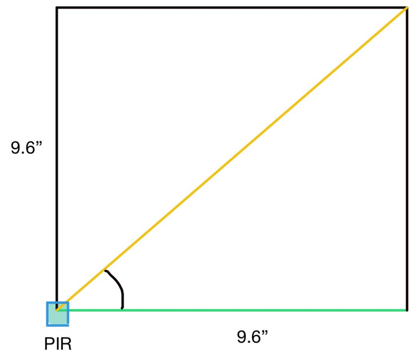


Figure 5: Top view

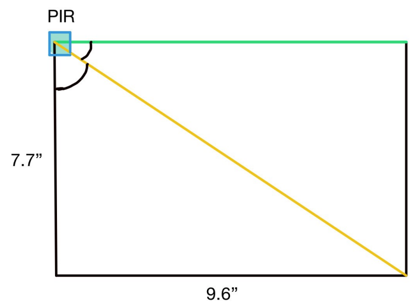


Figure 6: Side view

Based on these drawings, simple geometry can be used to solve for the angles on each side of the line of sight. In the top view, the angle on each side of the yellow line can be calculated by using $\Theta_1 = \Theta_2 = arctan(\frac{9.6}{9.6}) = 45^o$. For the side view the larger angle is found by using, $\Theta_3 = arctan(\frac{9.6}{7.7}) = 51.3^o$ and the smaller angle is found by using $\Theta_4 = arctan(\frac{7.7}{9.6}) = 38.7^o$. Based on these calculations, all four angles are well less than the 60^o limit we found from the datasheet and there will be no geometric restrictions for the PIR sensor.

3. Cost and Schedule

3.1. Cost Analysis

3.1.1. Parts Cost

Part	Quantity	Cost
ATmega328-A	1	$2.42
PIR Sensor	1	$3.95
Raspberry Pi (8GB) Model 5 Starter Kit	1	$169.95
Raspberry Pi Camera Module 3	1	$25.00
Bird Feeder	1	$25.99
Raspberry Pi Active Cooler	1	$5.00
PCB Enclosure	1	$27.10
Resistor 10K ohm	1	$0.10
Capacitor 1UF	1	$0.11
Linear Voltage Regulator IC Positive Fixed 1 Output 1A TO-252	1	$1.09
3x2 Programming Header	1	$1.75
CONN HEADER VERT 2POS	1	$0.10
Total Parts Cost per Unit		$262.56

3.1.2. *Labor Cost*

- Hourly Salary: $40
- Hours of work completed per week: 10 hours per week
- Total Weeks Worked: 16 weeks

$Total\ Hours\ =\ 10\ hr/week\ *\ 16\ weeks\ =\ 160\ hrs$

$Total\ Labor\ Cost\ per\ person\ =\ \$40/hr\ *\ 160\ hrs\ =\ \$6,400$

$Total\ Labor\ Cost\ =\ \$6,400/person\ *\ 3\ people\ =\ \$19,200$

3.1.3. Total Cost of product

$Total\ Cost\ =\ Total\ Labor\ Cost\ +\ Total\ Parts\ Cost\ =\ \$\ 19,453.51\ per\ Unit$

3.2. Schedule

Week	Tasks	Team Member
Week of 2/19	Finalize Project Proposal	Everyone
	Write Design Document	Everyone
	Breadboard test ESB32 and ultrasonic transducer	Kevin & John
	Order Raspberry Pi & Raspberry Camera	Colten
Week of 2/26	Design Review with Instructor and TAs	Everyone
	PCB Review	Everyone
	Unit test voltage regulators/design power subsystem	John
	Finalize PCB design for submission by Friday	Kevin
Week of 3/4	Start on Camera setup and implementation with Pi	Colten & Kevin
	Order PCB/Teamwork Evaluation	Everyone
	Test power subsystem on breadboard	John
	Make sure PCB has been ordered	Kevin
Week of 3/11 (Spring Break)	Start on data collection software on raspberry pi side	Colten & Kevin
	Get the waterproof container from machine shop	Kevin
	Buy Bird Feeder	Kevin
	Interface power subsystem with sensing subsystem and microcontroller, everything should be assembled except for the pi camera (½)	John
	Test PCB to see if it works	Kevin
Week of 3/18	Start on web server on Raspberry Pi and if time allows implement data on web server as well as video feed	Colten & Kevin
	Prepare presentation outline and visuals	Everyone
	Interface power subsystem with sensing subsystem and microcontroller, everything should be assembled except for the pi camera (2/2)	John

	Work and debug PCB schematic for next order	Kevin
	Order new PCB if needed	Kevin
Week of 3/25	Test and debug any problems with the Pi system	Colten & Kevin
	Test PCB to see if it works	Kevin
	Confirm any mechanical alterations to the birdhouse with machine shop	John
Week of 4/1	This week and next can be used if any further debugging is needed with Pi but would prefer to use this time for prep for presentation / help teammates with anything else.	Colten
	Connect raspberry pi to microcontroller output signal	John
	Test product with battery only (Aside from raspberry Pi)	Kevin
Week of 4/8	Fix any minor bugs	Everyone
	Final presentation preparation	Everyone
Week of 4/15	Mock Demo	Everyone
Week of 4/22	Final Demo	Everyone
Week of 4/29	Final Presentation	Everyone
	Submit Final Paper	Everyone

4. Discussion of Ethics and Safety:

With a project such as ours, ethics and safety are of the utmost importance for every step of the way. Addressing the ethical concerns there are two main concerns we have identified. First we are concerned that this project has similar products that are already commercially available, which means that we need to actively ensure that we are coming up with unique ideas that are firmly our own. The second ethical concern comes from the fact that there have been other past ECE 445 senior design projects that have similar subsystems to ours, and we want our project to be unquestionably unique. These two problems have similar solutions, and we have identified several ways to ensure that our project is ethically compliant with the IEEE code of ethics, mainly to comply with the tenet of "To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities." [1].

The physical birdhouse comes with two main safety concerns for our project. The first is water protection for our power system and electronics against inclement weather. This concern is addressed by our water proofing of our enclosure system. The second outdoors related concern is for squirrels and other unintended wildlife from harming the system. This problem is addressed by isolating our bird feeder and protecting all electric components from the environment with adequate security to prevent intrusions.

As a low voltage project, the safety concerns with this project are the same of most typical hardware design labs and we will observe all of the necessary precautions. The main concerns we have for safety in this project is following lab procedures during fabrication and testing. On the user end, the components will be contained in the bird feeder enclosure which means that they should be properly isolated from the electronics and power supply. Because the power supply will be contained within the bird feeder, we will need to make sure that there are no fire hazards in the storage of the supply [2].

The main safeguard we have against copying other products that are on market is to be active in our competitor research rather than being passive. This means we will be actively scouring the internet and other marketplaces for similar products and ensuring that the project we are completing is fully our own unique idea. We will address this during the design process by using our design notebooks to track our own research and creative processes. Tracking these dated and detailed entries will give definitive proof that we actively sought to ensure the individuality of our design process.

Overall, our project's safety and ethical concerns can be minimized and addressed through the use of active prevention techniques and engineering controls to ensure a safe, ethical project for both the fabrication team and the end user.

5. Citations

- IEEE ethics code

[1] I. R. Dutton, "Engineering code of ethics," in *IEEE Potentials*, vol. 9, no. 4, pp. 30-31, Dec. 1990, doi: 10.1109/45.65865.

- NEC 70 2023

[2] D. R. Crow and T. M. Crnko, "NFPA 70E," in *IEEE Industry Applications Magazine*, vol. 12, no. 1, pp. 43-49, Jan.-Feb. 2006, doi: 10.1109/MIA.2006.1578564.

- HC-SR501 Datasheet

[3] [1]“HC-SR501 PIR MOTION DETECTOR.” Available: https://www.mpja.com/download/31227sc.pdf

- L7800 Voltage regulators datasheets

[4] STMicroelectronics, “L7800 Series POSITIVE VOLTAGE REGULATORS,” L7805 datasheet, Nov. 2004 (accessed 2/7/2024).

- Raspberry Pi Datasheet

[5] “Raspberry Pi 5,” 2023. Available: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf

- ATmega328P Datasheet

[6] Atmel, “ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET,” 2015. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf